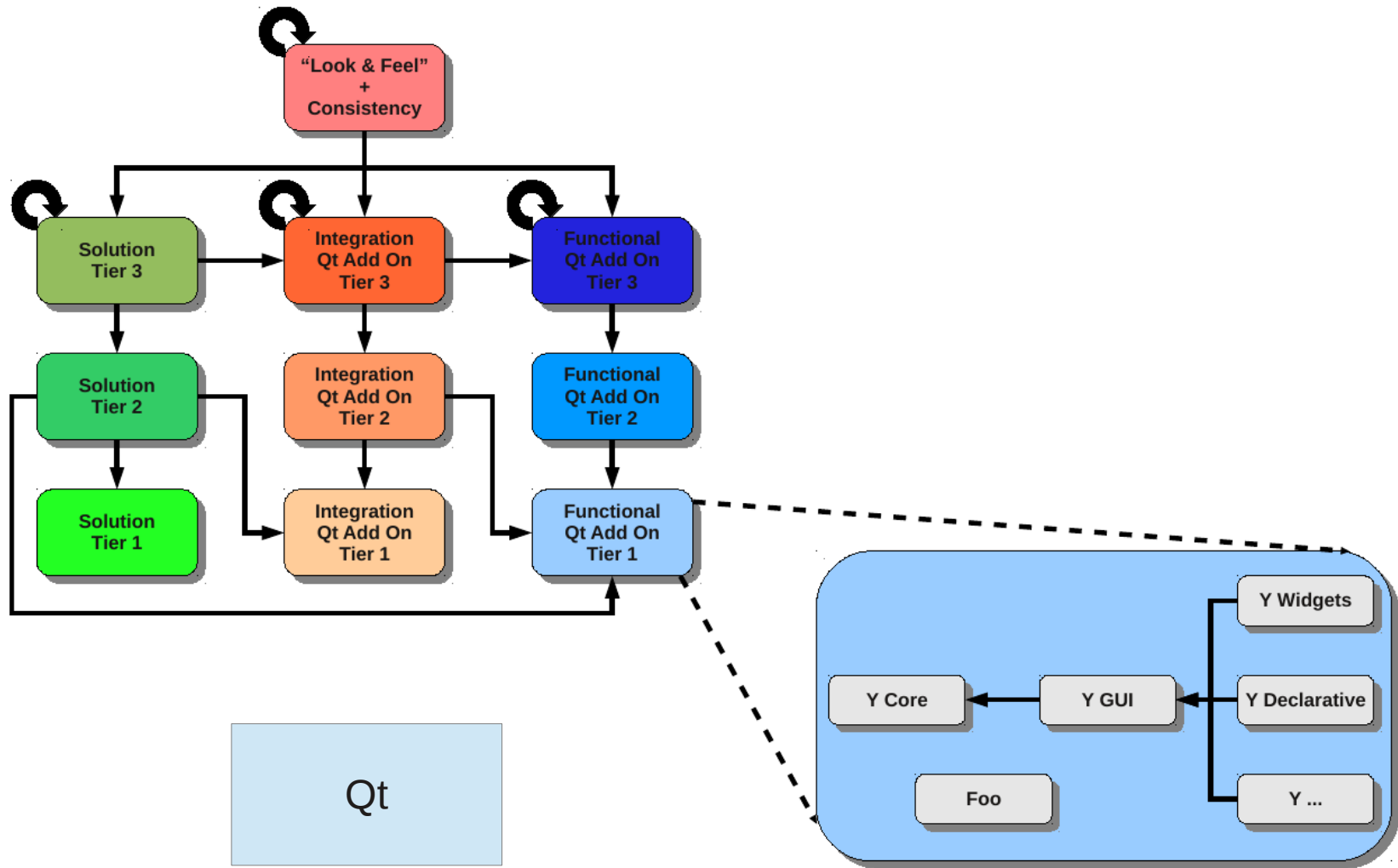


# Qt addons for everyone

KDE Frameworks 5

David Faure <[david.faure@kdab.com](mailto:david.faure@kdab.com)>

- Too many inter-dependencies in the API  
i18n → config → standarddirs → mainComponent → i18n  
KGlobal, central node for everything
- Too many dependencies on runtime bits  
desktop file → ksycoca → kbuildsycoca+kded → kdeinit
- No re-usability outside of KDE applications



Splitting based on external and Qt dependencies, for instance QtCore vs QtDBus, or QtCore (libkcore) vs bzip2 (libkarchive)

- Functional: no runtime dependencies
- Integration: optional runtime dependencies  
(for integration with OS/platform)
- Solutions: mandatory runtime dependencies

- Often requested in Qt: ZIP support
- Provided by KArchive
- ZIP, TAR, 7Zip archives
- Compression QIODevice – gzip, bzip2, xz
- Tier1, QtCore-only functional framework

- Complete spell-checking framework
- Core classes
  - Speller, Background checker
- Widgets
  - Spellcheck dialog, Config dialog, Highlighter...
- Plugins: aspell, hspell, hunspell, enchant
- Tier1, Integration framework (plugins)

- Hardware detection
  - Storage drives and volumes
  - Processor, Battery, Physical buttons, ...
  - Power management
  - Networking (status, interfaces)
  - In progress: bluetooth library (bluedevil)
- Tier 1, Integration framework
- Runtime deps: soliduiserver (encrypted partitions), daemons for power and networking.


- Job-based thread pool
- Features job dependencies, unlike QThreadPool
- done() signal (per-job), jobsDone() signal (global)
- Support for aborting, suspending...
- Tier1, QtCore-only functional framework



- Detecting when the user isn't using the computer
- Cross-platform
- Example use case: time tracking software
- Tier 1, Functional framework
- Currently depends on QtCore+QtWidgets

- Wrapper for DNS service discovery  
Zeroconf, Bonjour, Avahi
- Discover available services
- Announce availability of own service
- Tier1, QtCore+QtNetwork, functional framework

- Configuration framework
- INI format
- QSettings is suboptimal and deprecated
- XDG-compliant cascading directories
- Group-oriented API
- XML-based class generation
- Tier 1, QtCore functional framework  
(needs QLockFile, planned for Qt-5.1)

- Network-transparent Virtual File System
- Let the user browse and edit files the same way, locally and remotely 
- Asynchronous jobs, separate processes
- Many many protocols supported
- Tier 2 or 3, solution framework

- KCoreAddons (KJob...)
- XMLGUI
- Attica
- Phonon
- Soprano
- Nepomuk
- KCodecs (Encoding prober)
- KPlotting
- Notifications
- Bookmarks
- Item models
- KService
- KParts
- KDE i18n
- Plasma
- KDE Desktop Integration classes
- KDBusAddons (unique app...)
- KGuiAddons (Icon loading, color utils...)
- KAuth (mac/linux)
- Get Hot New Stuff
- Akonadi (PIM)



- New: `QStandardPaths` (replaces `KStandardDirs`)
- New: `QTemporaryDir` (replaces `KTempDir`)
- New: `QMimeType` (replaces `KMimeType`)
- New: `QEventLoopLocker` (replaces `KGlobal::ref/deref`)
- New: `QApplication::setApplicationDisplayName()`

- Improved: QTemporaryFile (replaces KTemporaryFile)
- Improved: QMimeData (replaces KUrl mimedata)
- Improved: qDebug (output, first steps for categories)
- Finally working: QUrl (replaces KUrl)



## QtCore

- Command-line arguments
- Global statics
- Safe file saving
- File locking
- URL path manipulation
- Locale and date/time (calendar support)

## QtWidgets

- QLineEdit: clear button, text squeezing, URL drops
- QTextEdit: placeholder text
- QMenu: titles, keyboard navigation
- QLabel: text squeezing
- QPushButton: delayed menu
- ...

- No technical distinction between a “pure Qt” application and a KDE application
- All apps are Qt apps, possibly using some frameworks on top
- Central repository for all Qt-based frameworks

<http://include.org/>

- Qt application developers use available frameworks rather than reinvent the wheel.
- Questions? (Other than a release date)